

Electric Car Sounds

By: Skril Studio



Thank you for purchasing Electric Car Sounds.

Contents

Package Includes.....	4
Importing E.C.S. asset package into your project.....	5
Adding compatibility for vehicle physics controller.....	5
The Sound Controller.....	6
Edit Settings.....	7
General Settings.....	7
Motor Audio Settings.....	8
Idle Sound.....	9
Static Sound.....	9
Wind Noise Audio Settings.....	10
Editor Settings.....	11
Setup for Custom Car Controller (your self made car physics controller).....	12
Setup for Unity Car.....	13
Setup for Edy's Vehicle Physics.....	14
Setup for Realistic Car Controller.....	15
Setup for Randomation Vehicle Physics.....	16
Setup for NWH1 and NWH2 Vehicle Physics.....	17
Setup for Motorbike Starter Kit.....	18
Setup for Vehicle Physics Pro.....	18
Setup for Unity Car Pro.....	19
Setup for Driver (Traffic System).....	19
Setup for iRDS (Intelligent Race Driver System).....	19
Setup for MS Vehicle System - Free.....	20
Setup for Realistic Car Kit.....	20
Setup for Universal Vehicle Controller.....	20
Setup for FGear Vehicle Physics.....	21
Setup for Realistic Car Controller Pro.....	21
Setup for Arcade Car Controller	22
Setup for Realistic Traffic Controller	22
Setup for Simple Motocross Physics	22

Setup for Simple Motorcycle Physics Pro23

Setup for Sim-Cade Vehicle Physics23

Setup for Motorbike Physics Tool.....23

Setup for Highroad Engine.....24

Setup for Arcade Bike Physics Pro.....24

Setup for Racing Game Starter Kit.....24

License.....25

Credits.....25

Package includes:

- Acceleration and deceleration wav files for exterior and interior camera views
- [optional] Various idle sounds, wind noise sounds and „static“ sounds for both exterior and interior camera views.
- Custom sound controller with nice looking user firendly inspector window.
- 36 prefabs (18 for exterior and 18 for interior camera views)
- 1 demo scene
- 266 wav files
- 36 prefabs
- 26 .unitypackages for vehicle physic controllers (containing prefabs and demo scenes):
 - Edy's Vehicle Physics
 - Realistic Car Controller
 - Realistic Car Controller Pro
 - Randomation Vehicle Physics
 - Unity Standard Assets Car
 - NWH Vehicle Physics
 - NWH2 Vehicle Physics
 - Motorbike Starter Kit
 - Vehicle Physics Pro
 - Unity Car Pro
 - Driver (Traffic System)
 - Intelligent Race Driver System (iRDS)
 - MS Vehicle System - Free
 - Realistic Car Kit
 - Universal Vehicle Controller
 - FGear Vehicle Physics
 - Arcade Car Controller
 - Realistic Traffic Controller
 - Simple Motocross Physics
 - Simple Motorcycle Physics Pro
 - Sim-Cade Vehicle Physics
 - Motorbike Physics Tool
 - Highroad Engine
 - Arcade Bike Physics Pro
 - Racing Game Starter Kit
- + an example compatibility for custom car controllers

There are total 266 Wav audio files in this package.

There are 36 prefabs in this package.

Importing R.E.S. asset package into your project

Import Electric Car Sounds into your project and later import the right *.unitypackage for the compatible car controller you're using in your project.

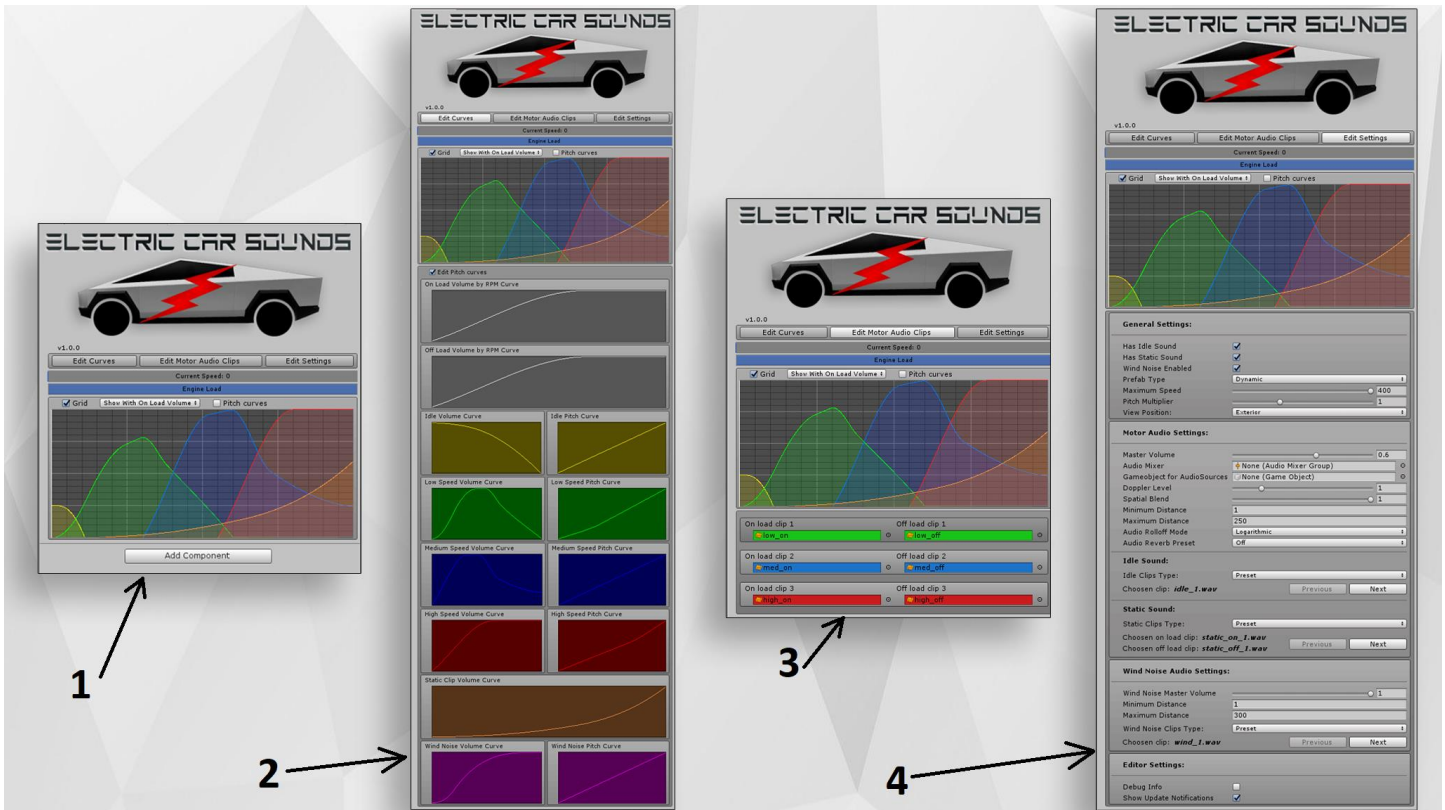
Adding compatibility for vehicle physics controller

When you're importing the asset into a project that is using a currently unsupported vehicle controller, you need to create your own script to set your vehicle physic controller's current speed and engine load to Electric Car Sounds prefabs. By engine load I mean to set acceleration pedal's value.

For exaples try loading a currently supported vehicle physics controllers *.unitypackage and take a look into the newly imported script. This will lead you into the right dirrection.

If you have problems adding compatibility to your vehicle physics controller, feel free to contact me about this and I will try to help you with a working script.

The Sound Controller



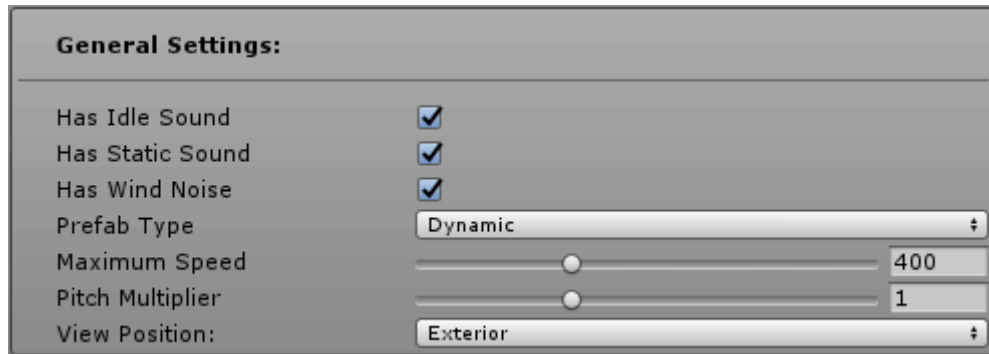
The asset comes with a sound controller that have a nice looking easy to use editor.

1. Default state with all volume curves being visible in the editor (this is allways visible in the inspector when one of the asset's prefab is selected)
2. Curve Editor, this is where you can edit ECS prefab's various values by clicking on a curve. A curve editor window will open, the changes you make to the curve will be visible immediately in the upper part if it is a curve that is visible there.
3. Edit Motor Audio Clips, this is where you can replace the electric motor's audio clips to yours if you would like to use your own clips.
4. Edit Settings, here you can change various settings of the prefab, enable/disable or change features and sound effects.

Edit Settings

General Settings:

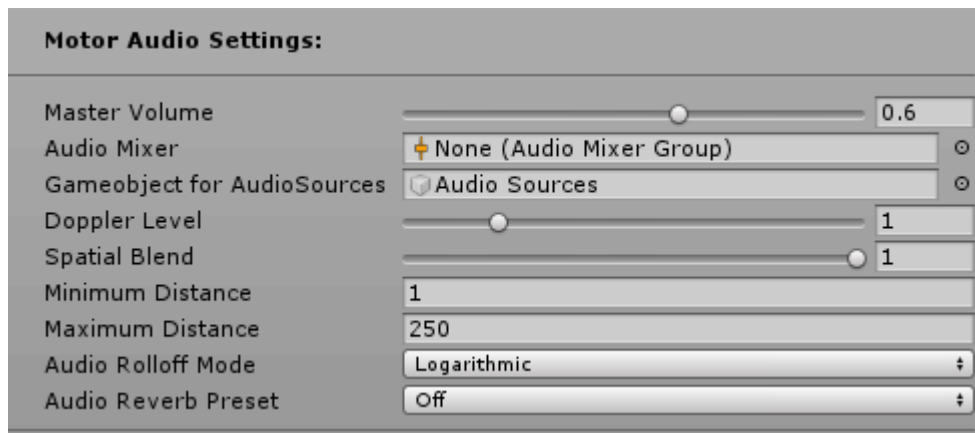
Change various settings, turn on or off features and sound effects.



- **Has idle Sound:** enable/disable idle sound. Enabling this feature will add an idle sound effect to the vehicle's sound.
- **Has Static Sound:** enable/disable static sound. Static sound effect's pitch does not change with the vehicle's current speed, only the volume changes with the vehicle's current speed. „*Pitch Multiplier*“ value can multiply the static sound effect's pitch.
- **Has Wind Noise:** Enabling wind noise will add a wind noise sound effect and a little detail to your vehicle's sound.
- **Prefab Type:** Sets how much sound clips will be used for the electric motor's sound in the prefab, these sound clips pitch and volume values are changing with the vehicle's current speed. It has two types: ***Simple*** and ***Dynamic***. In ***Simple*** mode only two sound clip is used. (1 on load, 1 off load sound clips) In ***Dynamic*** mode six sound clips are used. (3 on load, 3 off load sound clips).
- **Maximum Speed:** This should be equal with the car's maximum speed, but you can also set it by hand to a lower or higher value if you want to get different sound. If you set this to a lower value than your vehicle's max speed, it will reach faster the prefab's maximum level, and it will not change the pitch any more if your car is going faster than what you set for ***Maximum Speed*** in the ECS prefab. If you set this value higher than your vehicle's maximum speed the prefab will never reach its maximum level. (good for regular slow vehicles).
- **Pitch Multiplier:** Multiply all audio sources current pitch with the set value (except for wind noise).
- **View Position:** Choose in which camera view you will use this prefab. This will change the ***Idle sound***, ***Static Sound*** and ***Wind Noise*** sound clips made to exterior or to interior view sound clips. This does not change the motor sound clips to exterior/interior version.

Motor Audio Settings:

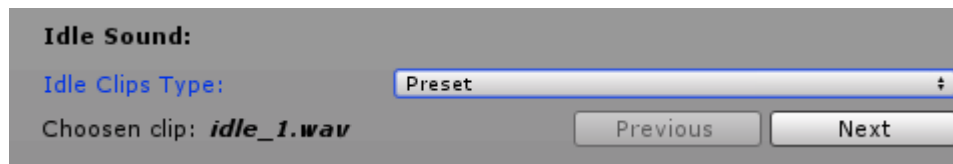
Sets the settings for all audio sources that will be created for this prefab.



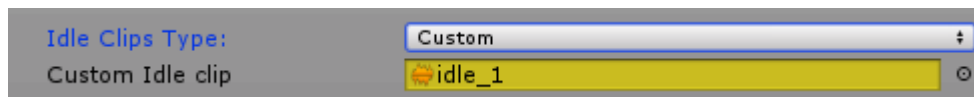
- **Master Volume:** The global volume of all sound clips within this prefab, except the wind noise.
- **Audio Mixer:** The Audio Mixer allows you to mix and master the vehicle prefabs with your other audio sources in the scene, you can apply effects to them in your audio mixer.
- **Gameobject for AudioSources:** This field can be left blank. Adding your own gameObject is good for setting the AudioSources' position in advance. If left blank a child gameObject will be automatically created in the prefab's position and all AudioSources will be created in this child gameObject.
- **Doppler Level:** Determines how much doppler effect will be applied to this audio source, if it is set to 0 then no effect is applied.
- **Spatial Blend:** Sets how much the AudioSources are affected by 3D spatialisation calculations (attenuation, doppler etc). 0.0f makes the sound full 2D, 1.0f makes it full 3D.
- **Minimum Distance:** Within the Min distance the AudioSources will cease to grow louder in volume.
- **Maximum Distance:** (Logarithmic rolloff) MaxDistance is the distance a sound stops attenuating at.
- **Audio Velocity Update Mode:** Describes when an AudioSource or AudioListener is updated. If you experience audio stuttering, try changing this value to Fixed mode or to Dynamic mode.
- **Audio Rolloff Mode:** Sets/Gets how the AudioSource attenuates over distance, this will be used for all audio sources even for wind noise too if it is enabled.
- **Audio Reverb Preset:** Adds a reverb effect to the vehicle's sound.

Idle Sound:

You can edit the Idle Sound's volume and pitch values in 'Edit Curves' tab by clicking on it's curve editor. (Only if it is enabled in General Settings). Preset audio clip changing buttons only work in Unity Editor, it will not work in a built game because the editor script is not building with the game.



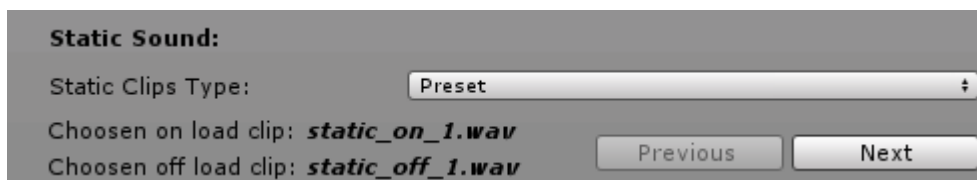
- **Idle Clips Type:** „Preset” sound clips are hard coded and can be chosen easily in the editor or set it to „Custom” type to use your own sound clips.
- **Chosen clip:** is the clip you choosed for the currently selected prefab. Can be changed with *Previous* and *Next* buttons.



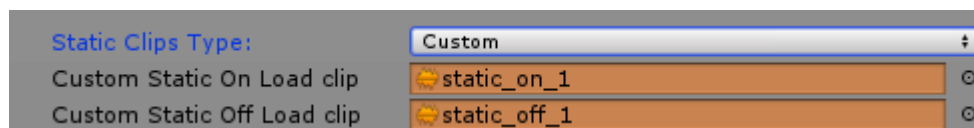
- **Custom Idle clip:** here you can add your own audio clips for idle sound.

Static Sound:

Static audios pitch does not change only just the volume. You can edit the Static Sound's volume value in 'Edit Curves' tab by clicking on it's curve editor. (Only if it is enabled in General Settings). Preset audio clip changing buttons only work in Unity Editor, it will not work in a built game because editor script are not building with the game.



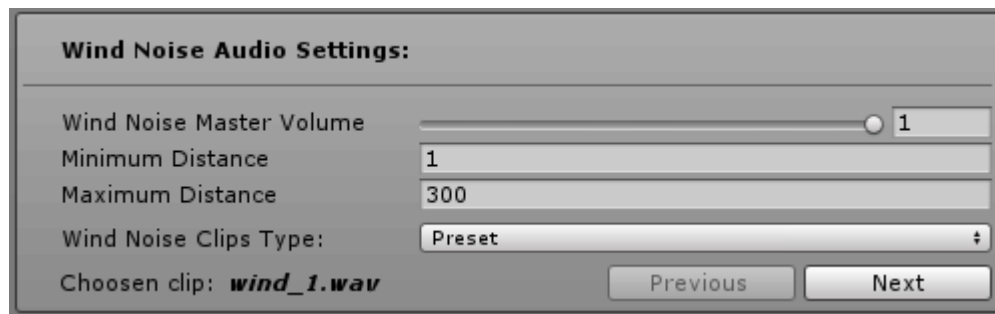
- **Static Clips Type:** „Preset” sound clips are hard coded and can be chosen easily in the editor or set it to „Custom” type to use your own sound clips.
- **Chosen on/off load clips:** is the clip you choosed for the currently selected prefab. Can be changed with *Previous* and *Next* buttons.



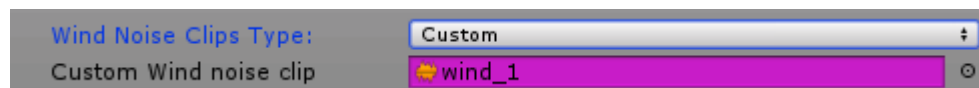
- **Custom Static On/Off Load clip:** here you can add your own audio clips for static sound. On Load clip is when the engine is on load/ accelerating, off load is when the engine in off load/decelerating.

Wind Noise Audio Settings:

You can edit the Wind Noise Sound's volume and pitch values in 'Edit Curves' tab by clicking on it's curve editor. Preset audio clip changing buttons only work in Unity Editor, it will not work in a built game because editor script is not building with the game.



- **Wind Noise Master Volume:** The global volume of wind noise within this prefab.
- **Minimum Distance:** Within the Min distance the wind noise will cease to grow louder in volume.
- **Maximum Distance:** (Logarithmic rolloff) MaxDistance is the distance a sound stops attenuating at.
- **Wind Noise Clips Type:** „Preset” sound clips are hard coded and can be choosen easily in the editor or set it to 'Custom' type to use your own sound clip.
- **Chosen clip:** is the clip you choosed for the currently selected prefab. Can be changed with *Previous* and *Next* buttons.



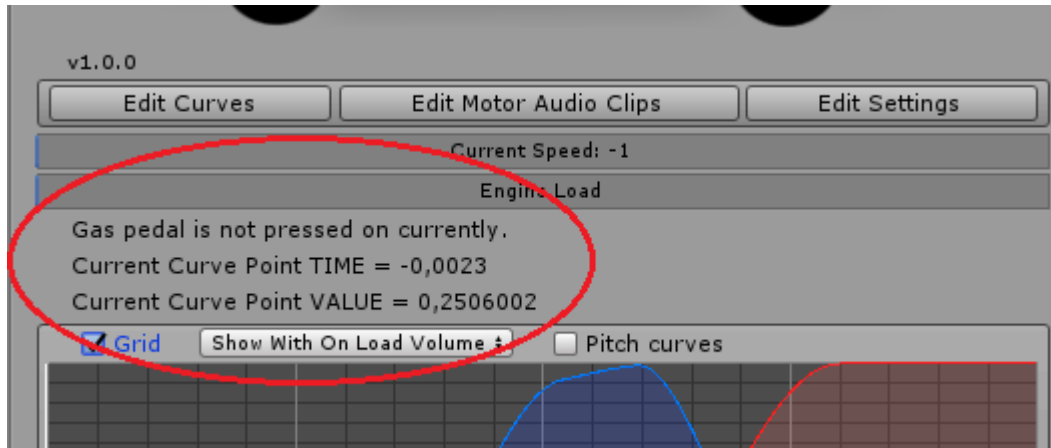
- **Custom Wind noise clip:** here you can add your own audio clips for wind noise sound.

Editor Settings:

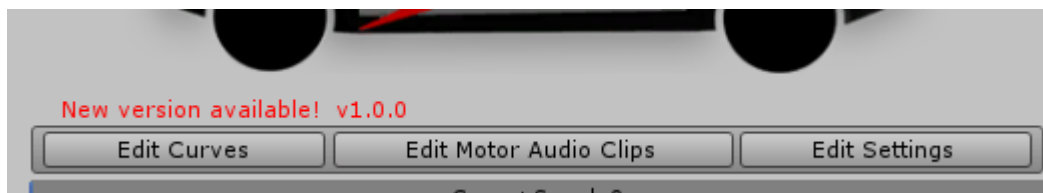
Change editor settings, turn on or off editor features.



- **Debug Info:** Showing detailed debug values in the inspector. See the image below:



- **Show Update Notifications:** Will show a notification message in the inspector under the asset's logo if a newer version of Electric Car Sounds is available for download. This can be turned off and it is saved globally for all Unity Projects. See the image below:



Setup for Custom Car Controller

This integration is made for custom car physics controllers, that you made by yourself.

The unitypackage can be found here:

\ElectricCarSounds\Assets_For_Vehicle_Controllers\CustomCarController\
CustomCarController_ECS.unitypackage

It contain a well commented example script for integration (*CustomCarController_ECS.cs*) which is a good quick start to connect your custom car controller with Electric Car Sounds. You need to modify this script in a way to make it work with your custom car controller. This script will convert values from your car controller to ECS.

You can find this script here after importing *CustomCarController_ECS.unitypackage*:

\ElectricCarSounds\Assets\Scripts\Demo Scene Scripts\CustomCarController

You need to connect the following values from your car controller to ECS:

car's current speed in km/h or mph -> **connects to** ecs.currentSpeed,
car's maximum speed in km/h or mph -> **connects to** ecs.maxTreshold,
car's gas pedal value, which is usually controlled by *Input.GetAxis("Vertical")*, if you controll it differently then edit the script according to that. 0 = gas pedal is released, more than 0 means gas pedal is pressed down.

You also need to rename CustomCarController to your car controller script's name.

The CustomCarController_ECS.cs script shows exactly how these values should be connected to ECS values. If everything is done right, ECS will became compatible with your custom car controller.

The unitypackage also contain demo scenes, where you can test all engine sounds by switching between them with UI buttons. You just need to drag and drop the pre-made gameobject named as "*ECS Prefabs- drop this into your car*" into your car, make sure this gameobject is positioned inside your car. If you modified the above mentioned script correctly, your car will have new engine sounds! :)

Setup for Unity Car

Youtube video url for this tutorial: <https://youtu.be/ev3M3jLe6ak>

Import Unity Standard Assets and Electric Car Sounds.

Now import **UnityStandardAssets_ECS.unpackage** from the following folder:

..\ElectricCarSounds\Assets For Vehicle Controllers\ UnityStandardAssets\..

This unpackage gives you prefabs and demo scenes for Unity Standard Assets car.

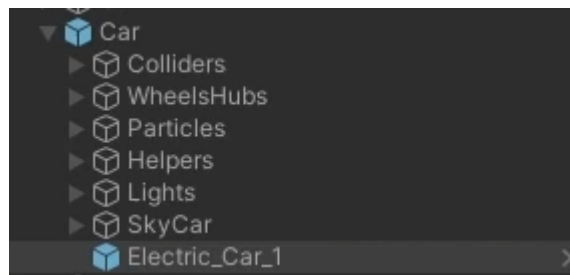
Open Unity Standard Assets_Car Sample Scene from the following folder:

../DemoScenes/Scenes/..

Drag and drop one of my prefab for Unity Car on your car's game object. You can find prefabs for Unity Standard Assets car in the following folder:

../ElectricCarSounds/Assets/Prefabs/StandardAssetsCar

Standard Assets car gameobject should look like this after you dropped one of my prefab on it :



That's all, now your Unity Standard Assets car have a cool engine sound, it's time to test it! Press Play button!



Setup for Edy's Vehicle Physics

Youtube video url for this tutorial: <https://youtu.be/Giykt42FUL8>

Import Edy's Vehicle Physics and Electric Car Sounds.

Now import **EVP_ECS.unitypackage** from the following folder:

../ElectricCarSounds/Assets For Vehicle Controllers/EVP/..

This unitypackage contain prefabs and demo a scene for Edy's Vehicle Physics.

How to use the prefabs made for EVP:

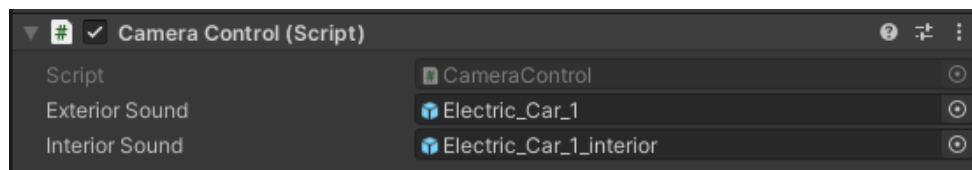
Open any Edy's Vehicle Physics Sample Scene.

Drag and drop one prefab made for EVP on your EVP car's game object. You can find prefabs for EVP vehicles in the following folder: **../ElectricCarSounds/Assets/Prefabs/RCC/..**

That's all, now your Edy's Vehicle Physics car have a cool electric engine sound, it's time to test it! Press the play button! 😊

These prefabs made for EVP are already containing the prefabs made for interior and exterior views and the camera controller script which let you choose between these with the default EVP camera changer button.

The camera controller script looks like this:

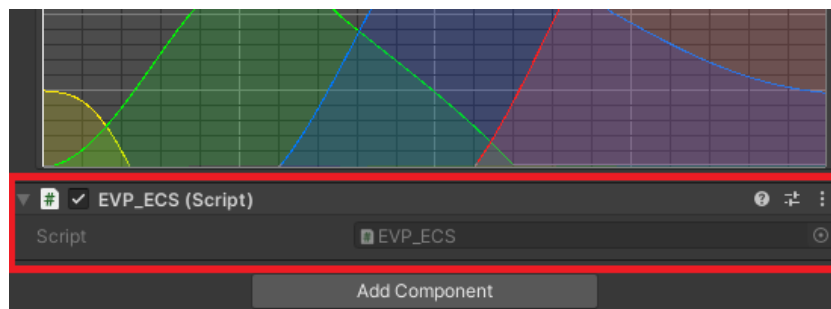


This script is used to detect which EVP camera is currently used and it will enable the right ECS prefab for the right camera view. (exterior or interior)

If you are making custom prefabs with your own sound files that have separate audio files for exterior and interior views, you can add those prefabs into this script and they will work with EVP's stock camera controller.

If you are making a custom prefab with sound files that are only made for exterior view, then you don't need to use this script. In this case you will have the same sound for exterior and interior camera views.

How to use this script? Create a new gameobject, add this script to the newly created gameobject and add your custom ECS exterior and interior view prefabs to the script. If you want to you can add your custom prefabs as a child object to this newly created gameobject for easier use and arrangement.



When you're making a custom ECS engine prefab, make sure that you have added EVP_ECS.cs script to your ECS prefab. This script translates EVP values to ECS.

Setup for Realistic Car Controller

Youtube video url for this tutorial: <https://youtu.be/MczyVia954M>

Import Realistic Car Controller and Electric Car Sounds.

Now import **RCC_ECS.unpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/RCC/..

This unpackage gives you prefabs and demo scenes for Realistic Car Controller.

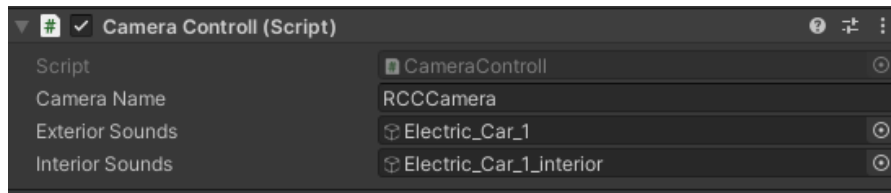
Open an Realistic Car Controller Scene, drag and drop one ECS prefab made for RCC on your RCC car's game object. You can find prefabs for RCC vehicles in the following folder:

../ElectricCarSounds/Assets/Prefabs/RCC/..

That's all, now your Realistic Car Controller vehicle have a cool electric engine sound, it's time to test it! Press the play button! 😊

These prefabs made for RCC are already containing the prefabs made for interior and exterior views and the camera controller script which let you choose between these with the default RCC camera changer button.

The camera controller script looks like this:



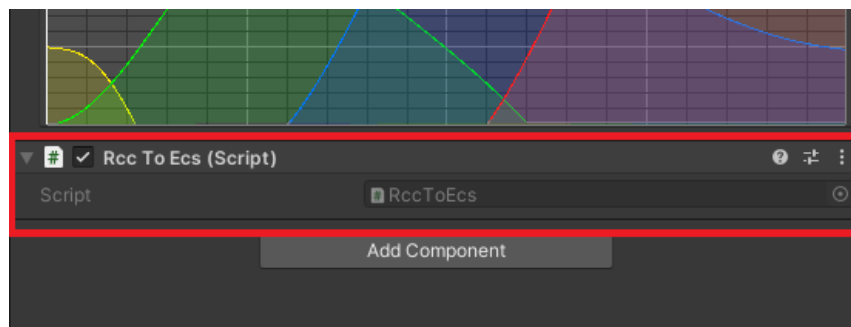
- **Camera Name:** RCC's camera name in the scene

This script is used to detect which RCC camera is currently used and it will enable the right ECS prefab for the right camera view. (exterior or interior)

If you are making custom prefabs with your own sound files that have separate audio files for exterior and interior views, you can add those prefabs into this script and they will work with RCC's stock camera controller.

If you are making a custom prefab with sound files that are only made for exterior view, then you don't need to use this script. In this case you will have the same sound for exterior and interior camera views.

How to use this script? Create a new gameobject, add this script to the newly created gameobject and add your custom ECS exterior and interior view prefabs to the script. If you want to you can add your custom prefabs as a child object to this newly created gameobject for easier use and arrangement.



When you're making a custom ECS engine prefab, make sure that you have added RCC_ECS.cs script to your ECS prefab. This script translates RCC values to ECS.

Setup for Randomation Vehicle Physics

Youtube video url for this tutorial: <https://youtu.be/rZy8DqDL1fl>

Import Randomation Vehicle Physics and Electric Car Sounds.

Now import **RVP2_ECS.unitypackage** from the following folder:

..\ElectricCarSounds\Assets For Vehicle Controllers\RVP2\..

This unitypackage gives you prefabs and demo scenes for Randomation Vehicle Physics.

Open any Randomation Vehicle Physics Sample Scene.

Spawn a car, drag and drop one of my prefab for Randomation Vehicle Physics on your car's game object. You can find prefabs for R.V.P. car in the following folder: ../ElectricCarSounds/Assets/Prefabs/RVP2

That's all, now your Randomation Vehicle Physics car have a cool engine sound, it's time to test it!😊

Setup for NWH1 and NWH2 Vehicle Physics

Youtube video url for NWH1 tutorial: <https://youtu.be/u71EeQvnnvRM>

Youtube video url for NWH2 tutorial: <https://youtu.be/8dzuRA2-Hxc>

Import NWH or NWH2 Vehicle Physics, set up to make it work and import Electric Car Sounds.

For NWH2 Vehicle Physics all the same steps will apply, the only difference is that instead of „NWH” text „NWH2” text is used.

Now import **NWH_ECS.unpackage** from the following folder:

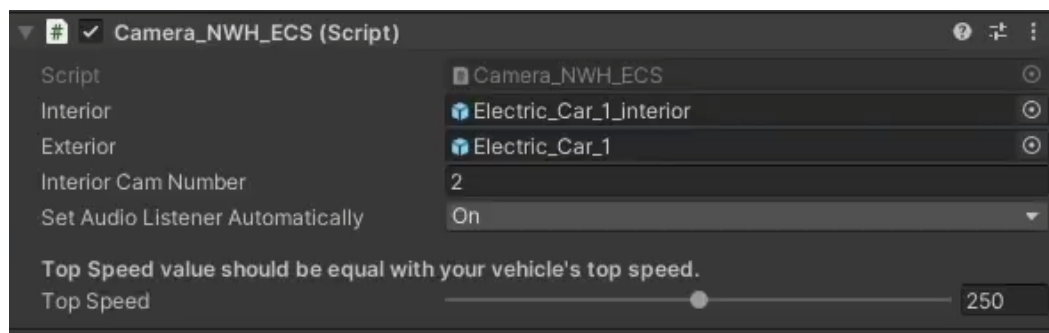
..\ElectricCarSounds\Assets For Vehicle Controllers\NWH\..

This unpackage gives you prefabs and demo scenes for NWH Vehicle Physics.

Open any NWH Vehicle Physics scene, drag and drop one ECS prefab made for NWH Vehicle Physics on your car's game object. You can find prefabs for NWH car in the following folder:

../ElectricCarSounds/Assets/Prefabs/ NWH

Camera NWH ECS controller script:



- Interior Cam Number: the order number of the car's interior camera gameobject (starting from 0)
- Set Audio Listener Automatically: if it is on, it will try to get the camera's audio source automatically
- Top Speed: it should be equal to your vehicle's top speed (minimum value is 100, the maximum is 400)

This prefabs are working with NWH's camera system, it will automatically switch between interior and exterior ECS prefabs.

That's all, now your NWH1 or NWH2 Vehicle Physics car have a cool engine sound! Enjoy it! ☺

Setup for Motorbike Starter Kit

Youtube video url for this tutorial: <https://youtu.be/rv5qjSZYHC4>

Firstly import Electric Car Sounds and Motorbike Starter Kit assets and later import

MSK_ECS.unitypackage from

../ElectricCarSounds/ Assets For Vehicle Controllers\MSK\.. folder.

This import prefabs, scripts and demo scenes for MSK and ECS.

Look for MSK-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/MSK/ folder.

Drag and drop one MSK-ECS prefab on your MSK bike. This prefabs contain a controller script that switch between helmet and exterior camera view prefabs.



HelmetCameraID: is the order number of helmet camera gameobject in your MSK bike.

Setup for Vehicle Physics Pro

Youtube video url for this tutorial: <https://youtu.be/mkTTXzMjnaU>

Firstly import Vehicle Physics Pro, Electric Car Sounds and later import **VPP_ECS.unitypackage** from **../ElectricCarSounds\Assets For Vehicle Controllers\VPP**

This import prefabs, scripts and demo scenes for VPP and ECS.

Look for VPP-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/VPP folder.

Drag and drop one VPP-RES prefab on your VPP vehicle. This prefabs contain a controller script that switch between interior and exterior camera view prefabs.

Setup for Unity Car Pro

Youtube video url for this tutorial: <https://youtu.be/9lX1vas2Mlw>

Firstly import Unity Car Pro, Electric Car Sounds and later import **UCP_ECS.unpackage** from **..\ElectricCarSounds\Assets For Vehicle Controllers\UnityCarPro**

This import prefabs, scripts and demo scenes for UCP and ECS.

Look for UCP-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/UCP folder.

Drag and drop one UCP-ECS prefab on your UCP vehicle. This prefabs contain a controller script that switch between interior and exterior camera view prefabs.

Setup for Driver (Traffic System)

Youtube video url for this tutorial: https://youtu.be/6_5a77tD_WM

Firstly import Driver (Traffic System), Electric Car Sounds and later import **Driver_ECS.unpackage** from **..\ElectricCarSounds\Assets For Vehicle Controllers\Driver**

This import prefabs, scripts and demo scenes for Driver and ECS.

Look for Driver-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/Driver folder.

Drag and drop the rightone Driver-ECS prefab on your Driver vehicle. There are separate prefabs for car and bikes. This prefabs contain a controller script that switch between interior/helmet and exterior camera view prefabs.

Setup for iRDS

Youtube video url for this tutorial: <https://youtu.be/obHK7qJ5e5M>

Firstly import iRDS, Electric Car Sounds and later import **iRDS-dll_ECS.unpackage** or import **iRDS-source_ECS.unpackage** if you are using the source version of iRDS from: **..\ElectricCarSounds\Assets For Vehicle Controllers\iRDS**

This will import prefabs, scripts and demo scenes for iRDS and ECS.

Look for iRDS-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/iRDS folder.

Drag and drop one iRDS-ECS prefab on your iRDS vehicle. These prefabs are containing a controller script that will switch between interior and exterior camera view prefabs.

Setup for MS Vehicle System - Free

Youtube video url for this tutorial: <https://youtu.be/l71c58VeqBo>

Firstly import MSV-Free, Electric Car Sounds and later import **MSVS-Free_ECS.unithpackage** from

..\ElectricCarSounds\Assets For Vehicle Controllers\MSVS-Free

This import prefabs, scripts and demo scenes for MSV-Free and ECS.

Look for MSV_Free-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/MSVS-Free folder.

Drag and drop one MSVS-ECS prefab on your MSVS vehicle. This prefabs contain a controller script that switch between interior and exterior camera view prefabs.

These prefabs and demo scene should work with MSVS paid version too after MSV_Free_ECS.cs and MSV_Camera_ECS.cs scripts got modified: rename „MSVehicleControllerFree” words to „MSVehicleController” in each above mentioned scripts.

Setup for Realistic Car Kit

Youtube video url for this tutorial: <https://youtu.be/62WOIKyE9zE>

Firstly import Realistic Car Kit, Electric Car Sounds and later import **RCK_ECS.unithpackage** from

..\ElectricCarSounds\Assets For Vehicle Controllers\RCK

This import prefabs, scripts and demo scenes for RCK and ECS.

Look for RCK-ECS prefabs in ../ElectricCarSouns/Assets/Prefabs/RCK folder.

Drag and drop one RCK-ECS prefab on your RCK vehicle. These prefabs are containing a controller script that will switch between interior and exterior camera view prefabs.

Setup for Universal Vehicle Controller

Youtube video url for this tutorial: <https://youtu.be/3mpxPRTtneM>

Import Universal Vehicle Controller "Without Dependencies" (make sure you done all required steps to make UVC work), then import Electric Car Sounds and later **UVC_ECS.unithpackage** from:

..\ElectricCarSounds\Assets For Vehicle Controllers\UVC

This unithpackage is containing demo scenes, scripts for demo scenes and prefabs for Universal Vehicle Controller. This prefabs are ready to use with Universal Vehicle Controller.

If you are going to use UVC with dependencies, then you need to do the following steps too:

UVC car will still play it's own engine sound, because it is using fmod and I not found any possibilities to turn it off inside the demo scenes without dissabling all other sounds too like surface sounds. If you found a better way or the asset got an update that will allow better sound customisation inside the demo scenes, let me know.

You can edit the fmod project to disable the stock engine sound or edit UVC's "CarSFX.cs" script by removing the following lines to stop sending rpm values to fmod, in this case the idle sound will still play in the background:

```
EngineEmitter.SetParameter (RPMID, Car.EngineRPM);
```

```
EngineEmitter.SetParameter (LoadID, Car.EngineLoad.Clamp (-1, 1));
```

At the time writing this tutorial, these line are 112 and 113. "CarSFX.cs" script can be found in:

..\UniversalVehicleController\Scripts\GamePlay\VehicleComponents\Car folder.

Setup for FGear Vehicle Physics

Youtube video url for this tutorial: <https://youtu.be/4oKwluwtFJM>

Import FGear Vehicle Physics, Electric Car Sounds and later **FGear_ECS.unitypackage** from:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/FGear

This unitypackage contain demo scenes, scripts for demo scenes and prefabs for FGear Vehicle Physics. These prefabs are ready to use with FGear Vehicle Physics.

Look for FGear-ECS prefabs in ../ElectricCarSounds/Assets/Prefabs/FGear folder.

Drag and drop one of these FGear-ECS prefab on your FGear vehicle to get electric car sounds.

You may also need to disable stock FGear engine sounds. To disable stock FGear engine sounds, in FGear car's "Effects" component set "Engine High Clip" and "Engine Low Clip" to "None".

- **Audio Velocity Update:** changing this can solve audio stuttering problems with car sounds, different FGear vehicle types may need different Audio Velocity Update mode.

Setup for Realistic Car Controller Pro

Youtube video url for this tutorial: https://youtu.be/q_g2Oguptfl

Import Realistic Car Controller Pro and Electric Car Sounds into your project.

Now import **RCCP_ECS.unitypackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/RCCP/..

This unitypackage contain a script, a prefab and demo scenes made for Realistic Car Controller Pro compatibility.

Drag and drop "RCCP ECS" prefab into an RCCP vehicle then drag and drop engine sound prefab into this prefab. You can add one prefab or both prefabs (exterior and interior prefabs of the same engine sound pack). When adding both engine sound prefabs made for exterior and interior views, the first one is always the exterior prefab.

For louder engine sound, increase exterior engine sound prefab's „Minimum Distance" value if you want louder engine sounds, it should be equal to the vehicle's size. Move the exterior engine prefab to a position where the vehicle will be at the center of „Minimum Distance"s sphere, and move the interior prefab where the vehicle's engine is positioned in reality.

RCCP to ECS controller script settings:

- **RCCP Camera Name:** your RCCP camera's name in your scene.

Setup for Arcade Car Controller

Youtube video url for this tutorial: <https://youtu.be/UjN-tNkl-yA>

Import Arcade Car Controller and Electric Car Sounds into your project.

Now import **ACC_ECS.unpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/ACC/..

This unpackage contain scripts, a prefab and a demo scene made for Arcade Car Controller compatibility.

Drag and drop “ACC to ECS” prefab into an ACC vehicle then drag and drop an engine sound prefab into this prefab.

Setup for Realistic Traffic Controller

Youtube video url for this tutorial: <https://youtu.be/WusD4o3fIH8>

Import Realistic Traffic Controller and Electric Car Sounds into your project.

Now import **RTC_ECS.unpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/RTC/..

This unpackage contain a script, a prefab and demo scenes made for Realistic Traffic Controller compatibility.

Drag and drop “RTC ECS Prefab” into a RTC vehicle prefab then drag and drop one ECS engine sound prefab into this prefab. Repeat this step for all RTC vehicle prefabs.

That’s all, now your Realistic Traffic Controller vehicles have cool electric engine sounds! Enjoy it! 😊

Setup for Simple Motocross Physics

Youtube video url for this tutorial: <https://www.youtube.com/watch?v=SjhPw1e6bjU>

Import Simple Motocross Physics and Electric Car Sounds into your project.

Now import **SMCP_ECS.unpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/SMCP/..

This unpackage contain a script, a prefab and demo scenes made for Simple Motocross Physics compatibility.

Drag and drop “ECS” prefab into a Simple Motocross Physics bike prefab then drag and drop one ECS engine sound prefab into „ECS” prefab.

That’s all, now your Simple Motocross Physics bikes have cool electric engine sounds! Enjoy it! 😊

Setup for Simple Motorcycle Physics Pro

Youtube video url for this tutorial: <https://www.youtube.com/watch?v=zwbKlDDaxIA>

Import Simple Motorcycle Physics Pro and Electric Car Sounds into your project.

Now import **SMP_ECS.unitypackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/SMP/..

This unitypackage contain a script, a prefab and demo scenes made for Simple Motorcycle Physics Pro compatibility.

Drag and drop “ECS” prefab into a Simple Motorcycle Physics Pro bike prefab then drag and drop one ECS engine sound prefab into „ECS” prefab.

That’s all, now your Simple Motorcycle Physics Pro bikes have cool electric engine sounds! Enjoy it! 😊

Setup for Sim-Cade Vehicle Physics

Youtube video url for this tutorial: <https://www.youtube.com/watch?v=SrsC5Gy8ZXo>

Import Sim-Cade Vehicle Physics and Electric Car Sounds into your project.

Now import **SCVP_ECS.unitypackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/SCVP/..

This unitypackage contain a script, a prefab and demo scenes made for Sim-Cade Vehicle Physics compatibility.

Drag and drop “ECS” prefab into a Sim-Cade Vehicle Physics car prefab then drag and drop one ECS engine sound prefab into „ECS” prefab.

That’s all, now your Sim-Cade Vehicle Physics cars have cool electric engine sounds! Enjoy it! 😊

Setup for Motorbike Physics Tool

Youtube video url for this tutorial: <https://www.youtube.com/watch?v=38jOD0mfsGc>

Import Motorbike Physics Tool and Electric Car Sounds into your project.

Now import **MPT_ECS.unitypackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/MPT/..

This unitypackage contain a scripts, a prefab and a demo scene made for Motorbike Physics Tool compatibility.

Drag and drop “MPT to ECS” prefab into a Motorbike Physics Tool car prefab then drag and drop one ECS engine sound prefab into „MPT to ECS” prefab.

That’s all, now your Motorbike Physics Tool bikes have cool electric engine sounds! Enjoy it! 😊

Setup for Highroad Engine

Youtube video url for this tutorial: <https://www.youtube.com/watch?v=h9i3K1x6LiI>

Import Highroad Engine and Electric Car Sounds into your project.

Now import **HE_ECS.unithpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/HE/..

This unithpackage contain a script, a prefab and demo scenes made for Highroad Engine compatibility.

Drag and drop "HE to ECS" prefab into HE vehicle then drag and drop an engine sound prefab into this prefab.

If the cars are too loud for you, edit the minDistance value in the HE_ECS.cs script at line 37 from 100 to 50 or 25 value.

That's all, now your Highroad Engine cars have cool electric engine sounds! Enjoy it! 😊

Setup for Arcade Bike Physics Pro

Youtube video url for this tutorial: <https://youtu.be/2U059aVAAO8>

Import Arcade Bike Physics Pro and Electric Car Sounds into your project.

Now import **ABPP_ECS.unithpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/ABPP/..

This unithpackage contain a script, a prefab and demo scenes made for Arcade Bike Physics Pro compatibility.

Drag and drop "ABPP to ECS" prefab into ABPP bike then drag and drop an engine sound prefab into this prefab.

That's all, now your Arcade Bike Physics Pro bikes have cool electric engine sounds! Enjoy it! 😊

Setup for Racing Game Starter Kit

Youtube video url for this tutorial: https://youtu.be/Hi5vRP_Wfd0

Import Racing Game Starter Kit and Electric Car Sounds into your project.

Now import **RGSK_ECS.unithpackage** from the following folder:

../ElectricCarSounds/Assets_For_Vehicle_Controllers/RGSK/..

This unithpackage contain a script, a prefab and demo scenes made for Racing Game Starter Kit compatibility.

Drag and drop "RGSK To ECS" prefab into RGSK vehicle then drag and drop an engine sound prefab into this prefab. Stock RGSK engine sound needs to be disabled by hand in car prefab, look for '**VehicleController**' script in your car prefab, then open **Sound** tab and set "**Engine Sound**" to "**None (Audio Clip)**".

That's all, now your Racing Game Starter Kit vehicles have cool electric engine sounds! Enjoy it! 😊

License

You can use this asset for unlimited games.

One license per seat for personal and commercial use.

You can't resell or redistribute the package or any single file from the package on any store!

THIS PRODUCT CAN ONLY BE USED IN VIDEO GAMES, YOU ARE NOT ALLOWED TO USE THESE SOUND FILES IN REAL VEHICLES OR TOYS. CONTACT ME FOR PRICING IF YOU WOULD LIKE TO USE THE SOUNDS IN PHYSICAL PRODUCTS.

Credits

Youtube Playlist:

https://www.youtube.com/watch?v=amTHBf67Gtw&list=PLYFtdNoo8S3h-tQgYL0rr_QUONP3xWcN6

Please write a review for my asset in the Unity Asset Store. Thank you!

Check out my other assets:

Realistic Engine Sounds 2: <https://bit.ly/3aZ50Ag>

Add-On packs for Realistic Engine Sounds 2: <https://bit.ly/2QYutD3>

Realistic Car Shaders: <http://bit.ly/39mVLXX>

UI SFX Mega Pack (2682 wav files): <https://bit.ly/40oKeCd>

Everything else: <http://bit.ly/2TBgK2z>

Made by: Skril Studio

Contact: info@skrilstudio.com

Website: <http://skrilstudio.com>

Facebook Page: <https://www.facebook.com/yugelmobile>

Speaking: English, Hungarian and Serbian.

May not be available to respond to messages during weekends and holidays.